

FIG. 1

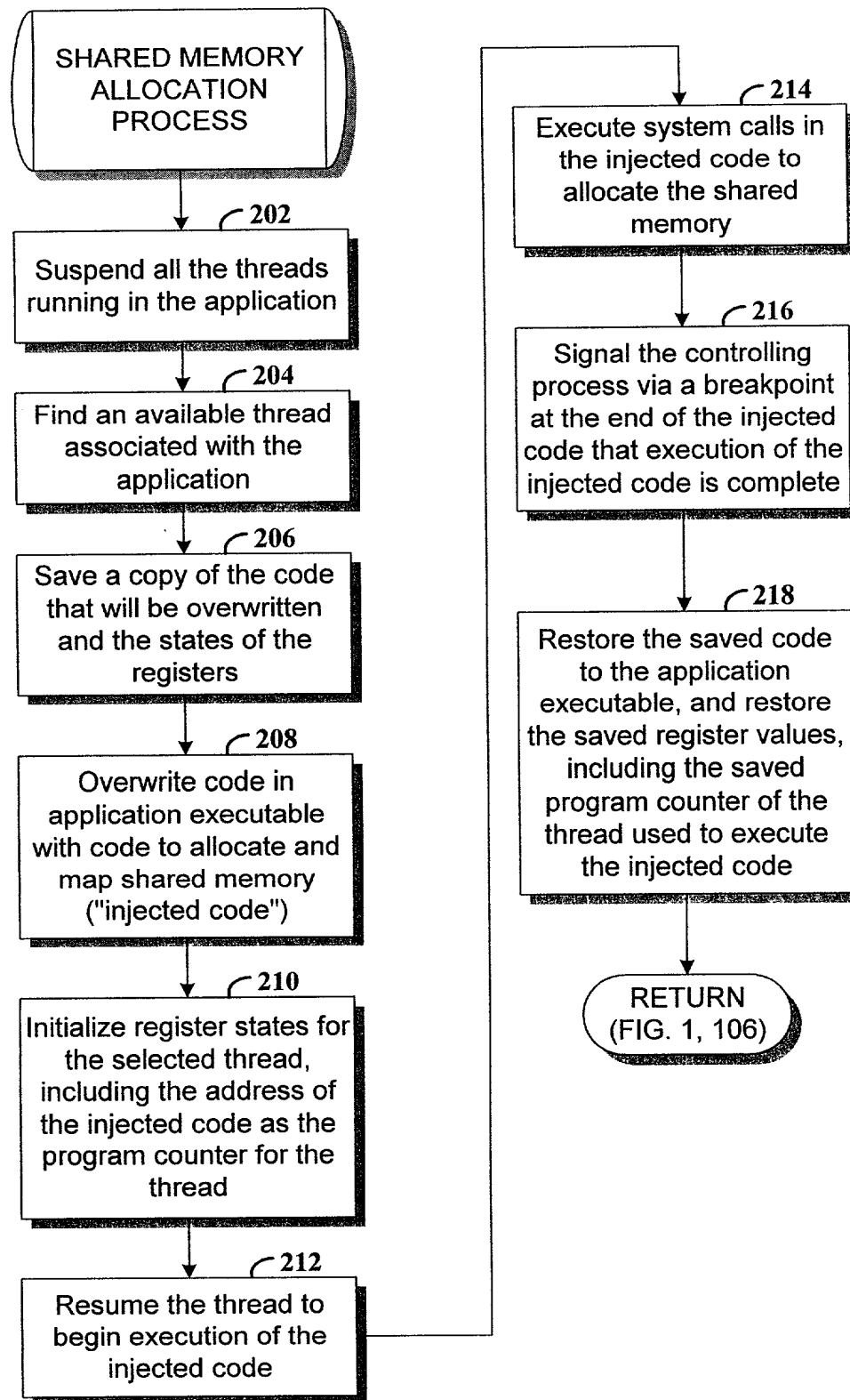


FIG. 2A

executable instrumentation program <u>302</u>	executable application <u>306</u>
instrumentation memory <u>304</u>	application memory <u>308</u>

FIG. 2B

executable instrumentation program <u>302</u>	executable application <u>306</u>
instrumentation memory <u>304</u>	code to allocate and map shared memory <u>312</u>
saved segment of code from executable application <u>310</u>	application memory <u>308</u>
shared memory <u>314</u>	

FIG. 2C

executable instrumentation program <u>302</u>	executable application <u>306</u>
instrumentation memory <u>304</u>	application memory <u>308</u>
shared memory <u>314</u>	

FIG. 2D

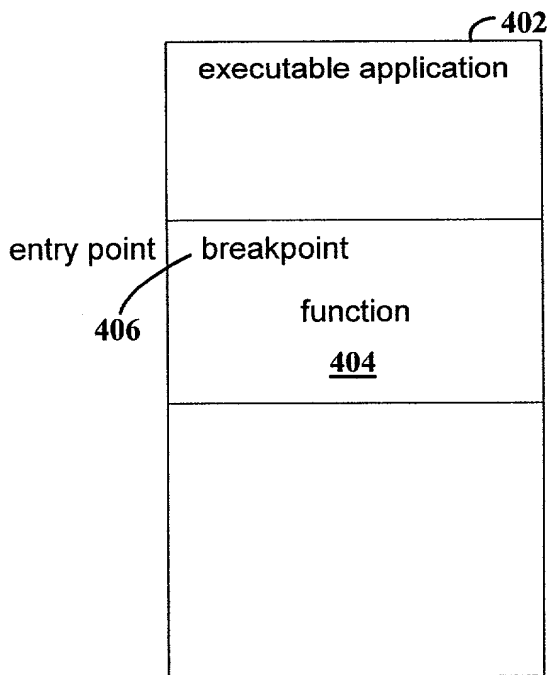


FIG. 3

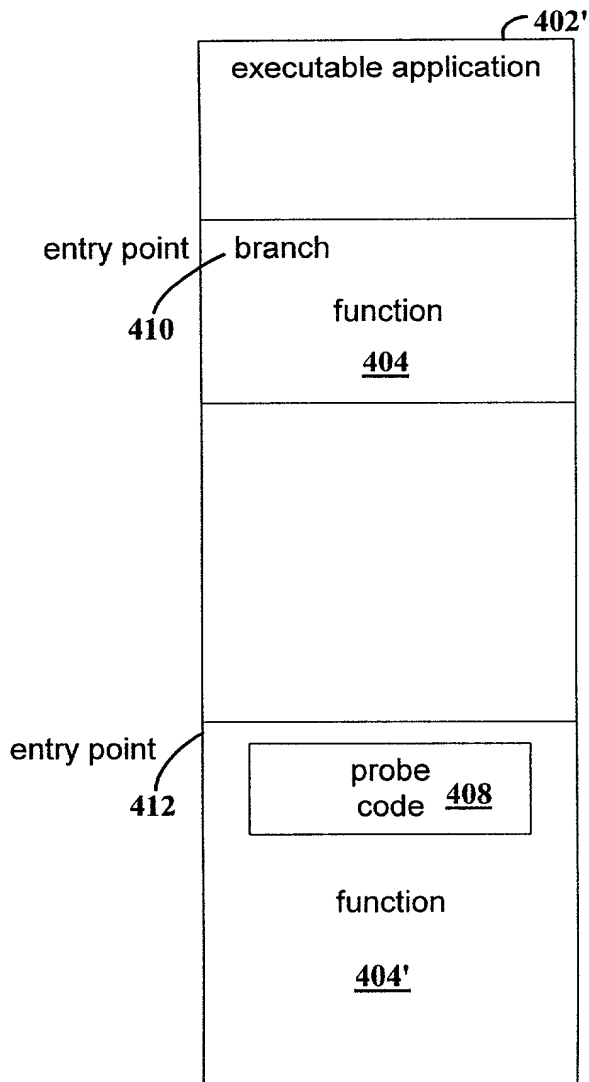


FIG. 4

FIG. 5 is a sequence diagram illustrating the process of instrumenting an application. The diagram shows three lifelines: 502 dynamic instrumentation, 504 executable application, and 506 instrumented function. The process begins with 502 injecting code (508) into 504. 504 then executes the injected code (510) and returns (514) to 502. 502 identifies functions and inserts breakpoints (516). 504 begins execution (518) and hits a breakpoint (522). 502 creates an instrumented function (524) and patches the original function's entry point with a branch to the instrumented function (526). 504 branches to the instrumented function (528), where probe code and the original function code are executed (530). Finally, 506 returns (532) to 504, and the process continues (indicated by a vertical arrow and ellipsis).

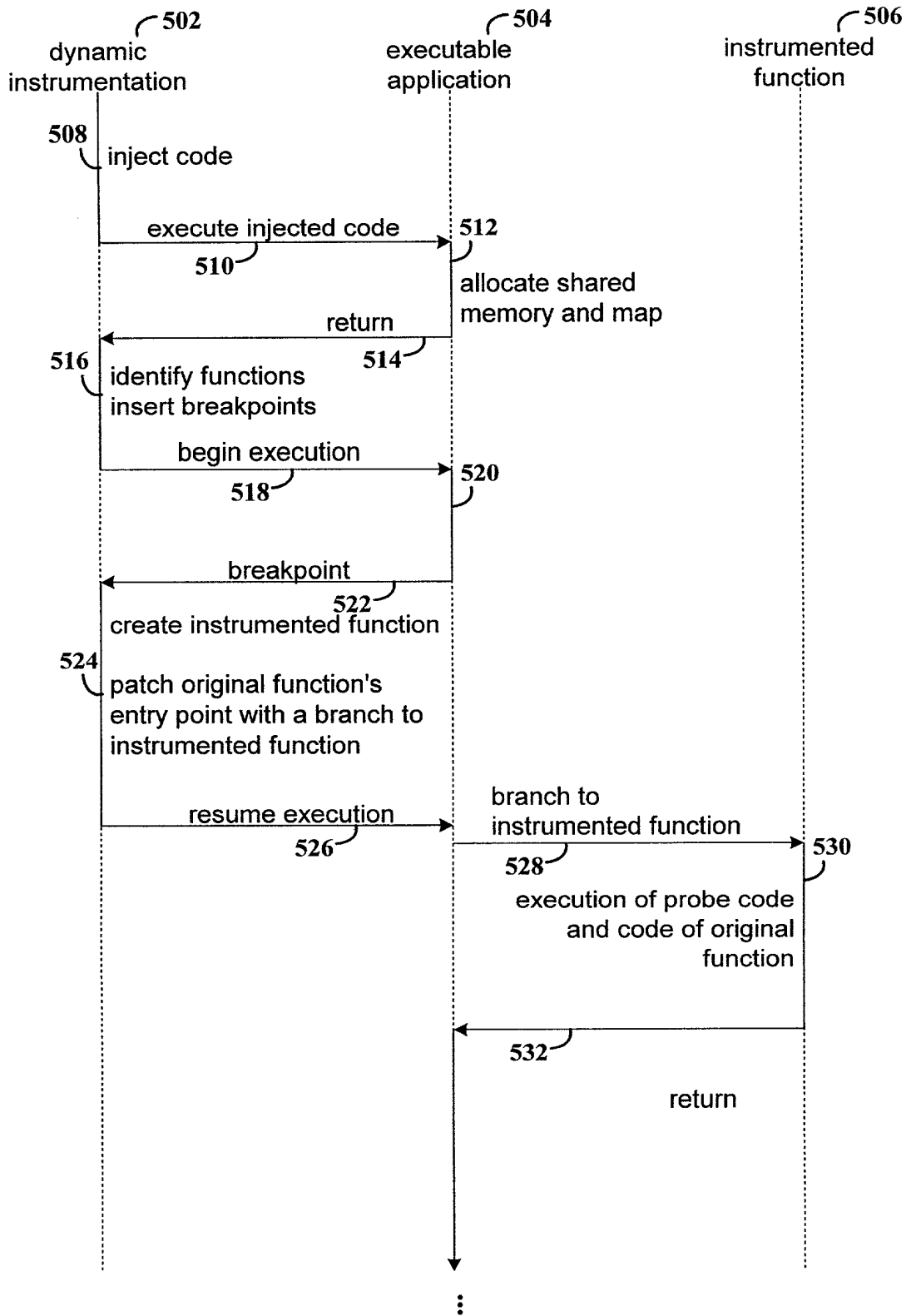


FIG. 5